**Лабораторная работа N12.**

**Реализация алгоритмов растровой графики для заполнения сплошных областей.**

Цель работы: Закрепить лекционный материал по изучению базовых алгоритмов компьютерной графики - алгоритмов закраски.

Краткие теоретические сведения.

Возможность представления сплошных областей - это одна из уникальных характеристик растрового дисплея. Все возможные методы такого представления можно разделить на две категории: растровая развертка и затравочное заполнение.

В первой в порядке сканирования строк определяют лежит ли точка внутри области. Такие алгоритмы применимы и к векторным дисплеям для штриховки контуров.

Во второй предполагают известной некоторую точку внутри замкнутого контура. Ее называют затравочной и ищут соседние с ней точки. Если найденная точка не внутри контура, значит она принадлежит границе, если внутри - она становится новой затравочной и поиск продолжается. Подобные алгоритмы применимы только к растровым дисплеям.

Многие замкнутые контуры либо многоугольники, либо их можно аппроксимировать простыми многоугольниками. Поэтому далее будем говорить о заполнении многоугольников.

***Растровая развертка многоугольников.***

Под многоугольником будем понимать фигуру на плоскости, ограниченную не самопересекающейся замкнутой ломаной. Основная идея растровой развертки многоугольников заключается в заполнении лишь одной строки в каждый данный момент.

Алгоритмы, работающие по этому принципу называют алгоритмами построчного сканирования. Простым способом выделения пикселов сканирующей строки попадающих внутрь многоугольника, является поочередная проверка всех пикселов строки растра (экрана). Но обычно большая часть пикселов лежит вне многоугольника, поэтому данный метод неэффективен во времени. Его можно несколько улучшить, помещая многоугольник внутрь минимально объемлющего прямоугольника со сторонами, параллельными сторонам растра, и анализируя лишь точки внутри этого прямоугольника. Но можно найти более эффективный метод, если допустить с большой долей вероятности, что соседние пикселы в строке (столбце) имеют одинаковые характеристики. Это допущение основывается на понятии пространственной когерентности: при перемещении от точки к точке или от одной сканирующей строки к соседней рассматриваемый многоугольник в большинстве случаев остается постоянным. Можно искать только те точки, где когерентность строк(столбцов) нарушена.

Пусть многоугольник задан набором своих вершин, причем соседние точки в наборе - его смежные вершины. Тогда на каждой сканирующей строке внутренние пикселы можно закрасить так:

1.Найти точки пересечения сканирующей строки со всеми ребрами многоугольника, кроме горизонтальных.

2.Отсортировать найденные точки по возрастанию координаты х.

3.Закрасить все пикселы между парами точек пересечений.

Алгоритм работает только в случае существенных пересечений и четного количества точек пересечений. Нечетное количество получается при пересечении сканирующей строки с локальными минимумами и максимумами, когда Y - координаты предыдущей и последующей вершин больше (меньше) чем у рассматриваемой. В этом случае такие пересечения учитываются как два. После такой корректировки алгоритм будет работать правильно при любой форме многоугольника.

Для выпуклых многоугольников алгоритм можно упростить и повысить его эффективность. Границу выпуклого многоугольника разбивают на две ломаные: "левую" и "правую" и, возможно, два ребра "верхнее" и "нижнее", тогда каждая ломаная будет иметь ровно одно пересечение с каждой сканирующей прямой. Используя алгоритм Брезенхема и одновременно генерируя растровое представление обеих ломаных, найдем левый и правый пикселы границы многоугольника на каждой сканирующей прямой. Последовательно заполняя интервалы между этими пикселами от верхней строки к нижней, получим итоговую растровую развертку выпуклого многоугольника.

***Алгоритмы заполнения с затравкой.***

Ищут и закрашивают связную компоненту области, содержащую затравочный пиксел. Под связностью понимают 4-х или 8-и связность в зависимости от постановки задачи. Каждый из пикселов четырехсвязной области достижим из любого другого пиксела этой области с помощью последовательности перемещений на пиксел из четырех элементного набора горизонтальных и вертикальных направлений, восьмисвязной - из восьмиэлементного (добавляются еще и диагональные направления). Алгоритмы для восьмисвязных областей применимы к четырехсвязным, но не наоборот.

Процесс заполнения областей иногда называют заливкой, так как можно представить, что в точке затравки находится источник, заливающий всю область определенным цветом. Область может быть определена внутренне (все внутренние пикселы – одного цвета, внешние -другого) и гранично (все пикселы на границе или вне области - одного цвета, не такого, как внутри). Далее будет рассмотрен гранично-заполняющий алгоритм для четырехсвязных областей (соответствующие внутренне-заполняющие алгоритмы можно получить аналогичным путем, а алгоритмы для четырехсвязной области легко переделать в алгоритмы для восьмисвязной).

Построчный алгоритм заливки.

Применим идею построчного сканирования и используем стек для хранения одного затравочного пиксела для любого непрерывного интервала на сканирующей строке. Непрерывный интервал – это группа примыкающих друг к другу пикселов, ограниченная уже закрашенными или граничными пикселами. Если непрерывный интервал пикселов принадлежит внутренней части области, пикселы на и под этим интервалом либо граничные, либо тоже находятся внутри области и могут служит затравочными для своих строк. Поэтому можно предложить следующую схему заполнения:

Описание алгоpитма:

Дано:

Pop(x,y) -процедура извлечения из стека координат (x,y) очередного пиксела

Push(x,y) - процедурапомещения в стек координат (x,y) очередного пиксела

Нарисовать\_точку(x,y,col) - процедура для подкрашивания цветом col пиксела с координатами(x,y) с(x,y) - цвет пиксела с координатами (x,y)

cb - цвет для подкрашивания пикселов границы области

ci - цвет для подкрашивания пикселов внутри области

(х0,y0) - координаты затравочного пиксела

Получить: Перекрасить сплошнуюобластьс данным затравочным пикселом и цветом границы в цвет ci.

1.Инициализируем стек:Push(x0,y0).

2.Пока стек не пуст выполнить:

начало

Извлекаем пиксел из стека и закрашиваем его в цвет области:

Pop(x,y);

Нарисовать\_точку(x,y,ci);

3.Присвоить: xw=x; x=x+1.

4.Пока c(x,y) # cb заполняем интервал справа от затравки:

Нарисовать\_точку(x,y,ci);

x=x+1

по окончанию идти к 5.

5.Присвоить: xr=x-1; x=xw; x=x-1.

6.Пока c(x,y) # cb заполняем интервал слева от затравки:

Нарисовать\_точку(x,y,ci);

x=x-1

по окончанию идти к 7.

7.Сохраняем крайний слева пиксел: xl=x+1;

8.Проверяем строки ниже и выше данной,если там есть еще неза-

полненные пикселы находим затравку начиная с левого края:

Для j от -1 до 2 с шагом 3 выполнить:

начало

x=xl; y=y+j;

Пока x <= xr ищем затравку на строке ниже(выше):

начало

fl="ложь";

Пока ( c(x,y)#cb and c(x,y)#ci and x<xr ) заполнить точ-

ки внутри:

начало

Увеличить х на единицу.

Если (not fl) = "истина" то присвоить fl="истина".

конец

Если fl="истина" то крайний справа пиксел - в стек:

начало

Если ( x=xr and c(x,y)#cb and c(x,y)#ci ) то Push(x,y)

иначе Push(x-1,y).

Присвоить fl="ложь".

конец

Продолжаем проверку,если интервал был прерван:

Присвоить xb=x.

Пока (c(x,y)=cb or c(x,y)=ci and x<xr) увеличивать x на 1.

Если x=xb то присвоить: x=x+1.

конец

конец

конец пункта 2.

6.Закончить.

Алгоритм правильно заполняет любую область, справляется с внутренними дырами и зубцами на границе. Используется при закраске, когда известна хотя бы одна точка внутри необходимой для закрашивания области.

**Задание на лабораторную работу:**

1. Написать на языке PASCAL программу, реализующую алгоритм заливки многоугольника любой формы.

2. Проверить правильность работы программы, нарисовав с помощью функций модуля GRAPH невыпуклый многоугольник(с дырами внутри) и закрасив его заданным цветом, указав координаты принадлежащей многоугольнику точки.

3. Написать и отладить программу, реализующую алгоритм построчного заполнения выпуклого многоугольника, заданного координатами вершин и цветом границы.

**Дополнительное задание:**

Пункт 2 задания выполнить с помощью программ работы с мышью.

***Вопросы для самоподготовки.***

1. Каким образом можно заполнить многоугольник.

2. Какие проблемы возникают при заполнении невыпуклого многоугольника, когда не известна ни одна точка, ему принадлежащая.

3. Как при заполнении сплошных областей можно использовать стек.

4. Что такое когерентность по строкам и столбцам и как, используя это свойство закрашиваемых областей, можно улучшить алгоритм закраски, сделав его более эффективным.

5. Чем отличаются алгоритмы построчного сканирования от алгоритмов заливки.

Литература:

1.Фоли Дж., Ван Дэм А. Основы интерактивной машинной графики. - М.: Мир, 1985, кн.2

2.Роджерс Д. Алгоритмические основы машинной графики.-М.: Мир, 1989

3.Шикин Е.В., Боресков А.В., Зайцев А.А. Начала компьютерной графики. - М.: ДИАЛОГ-МИФИ,1993

4.Шикин Е.В., Боресков А.В. Компьютерная графика. - М.: ДИАЛОГ-МИФИ, 1995

5.Шикин Е.В., Боресков А.В. Компьютерная графика. Полигональные модели. - М.: ДИАЛОГ-МИФИ,2000.-464с.